CS M148 Final Report

Ryan Riahi

May 2021

1 Executive Summary

In this report I will dive into the work that I did in building a model to try and predict whether patients are at risk for having a stroke. I used a dataset containing roughly 5000 (hopefully) random samples of patients information along with whether or not they have had a stroke. The data was already well structured and luckily did not need too much work. I first implemented a simple pipeline and then split the dataset into a train and test set and then over sampled the number of positives in attempt to balance the dataset. Lastly, I ran numerous different models on the data. The methods and models I used were logisitic regression, principal component analysis, bagging, neural network, and random forrest. I also implemented cross validation to try and get the most accurate results on how well my model performed.

2 Introduction

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. The total cost of stroke in the US was 103.5 billion dollars according to 2016 US dollar values. 68.5 billion dollars or 66% of total cost was accounted for by indirect cost from underemployment and premature death. Age groups 45-64 years accounted for the greatest stroke related direct cost.

In this project I attempt to predict whether a patient is likely to get a stroke based on the input parameters available. I use features such as gender, age, various medical conditions, and smoking status to build a model that attempts to predict if a person is likely to experience a stroke event in the near future.

Predicting stroke risk is a very sought after ability due to the worldwide effects of the issue. There have been previous findings that KNN and decision trees are good for predicting strokes. I found that, in fact, convolutional neural networks and Random Forrest decision trees are the best models to use.

3 Methodology

After loading the dataset, the first important thing I needed to get done was to pipeline the data. For this dataset, I used a simple pipeline because I felt that was all that was necessary, as the data was already well put together. The only column with missing values was the bmi feature, thus, I decided to use a median imputation strategy in order to handle the null values. For all of the categorical data I used a one hot encoder. The reason for this was that none of the categorical variables had many categories to them, so one hot encoding them seemed like a good choice. Lastly, I added one new feature which was based off bmi index along with glucose intake and then I put all the numerical features through a standard scalar.

After having the pipe lined data, I then built my various models. I would run the model with the base parameters and with my training and testing data. I would then look at the results and performance of my model and then try to optimize the model by tweaking various parameters until I got the best possible results I could. My main metric was F1 score, but overall it was all four metrics that were important to me.

4 Results

Now I will examine the results of each model and its performance. Plain logistic regression was definitively the worst model, and this was to be expected. However, the model wasn't even half bad, with an F1 score of 0.78. Logistic regression combined with principal component analysis was marginally better than normal logistic regression. As a result of this, I started using the PCA'd version of the dataset for the remainder of my models. My PCA was built to encompass 99% of the variance in the data. For both of the logistic models I used an l2 loss function along with a newton-cg solver. This didn't end up making much of a difference, however. I got around the same performance with the various solvers.

The bagging classifier was a significant improvement from logistic regression. The ROC curve improved along with other metrics, such as F1 which improved to 0.875. I used 10 estimators for the bagging, although this didn't make too much of a difference either. Bagging is definitely a good model for this kind of data.

After bagging I moved on the using a neural net. Specifically, I used sklearn's MLP Classifier (multi layer perceptron). This was the best model I had run so far and achieved amazing results. The ROC curve was almost a straight L line. The accuracy was 0.97, F1 was 0.97 and recall was 1.0! This result took me by surprise as to how the model could be performing so well. conclusion I came to was that potentially the upsampling allowed the model to have an extremely high recall score. So, in reality, I would imagine the recall score would be slightly lower. For the neural net I used a lbfgs solver along with a low alpha and a large number of iterations, which gave me optimal performance results.

Lastly, my best performing model was random forrest which managed to achieve results even better than the neural net! I suspect the reasoning behind this is the same as the reason that the neural net did so well on this dataset. With 500 estimators for random forrest, I managed an AUC that is essentially equal to 1.

5 Discussion

A big takeaway from the results is that these models performed really well, especially when compared relatively to previous models built for this kind of prediction. A big reason I think that the models performed so well is that the oversampling allowed the model to predict positives with ease at the cost of precision. I would say that the UCLA hospital should be cautious when employing my models, but I think the model can give a fairly strong prediction that someone will be having a stroke, due to the fact that the recall rate was so small. I believe this recall rate can be trusted, however, because the cross validation scores were very strong, especially with recall, on both the neural net and random forrest.

6 Conclusion

In conclusion, I believe that the models I have created work really well on the given data. There is no way to be sure that the data that I based everything off of is truly random and an accurate sample of the population as a whole; however, I do think the models provide a benefit regardless. This is also in spite of the fact that our dataset was highly imbalanced and that much work was needed to make it work. The biggest success from this is the recall score. In predicting stroke, I would say that recall is the most important. Being able to trust positive predictions to encompass true positives is huge. Even if there is a false positive, the worst case is that that person is extra weary and has some extra inconveniences. The worst case is a false negative, which does nothing to help alleviate the problems of stroke throughout the world. That is why I believe that these models can genuinely have a large positive impact on the health of numerous individuals.

7 References

`https://scikit-learn.org/stable/`
`https://numpy.org/`
`https://pandas.pydata.org/`